

Báo cáo cuối kỳ





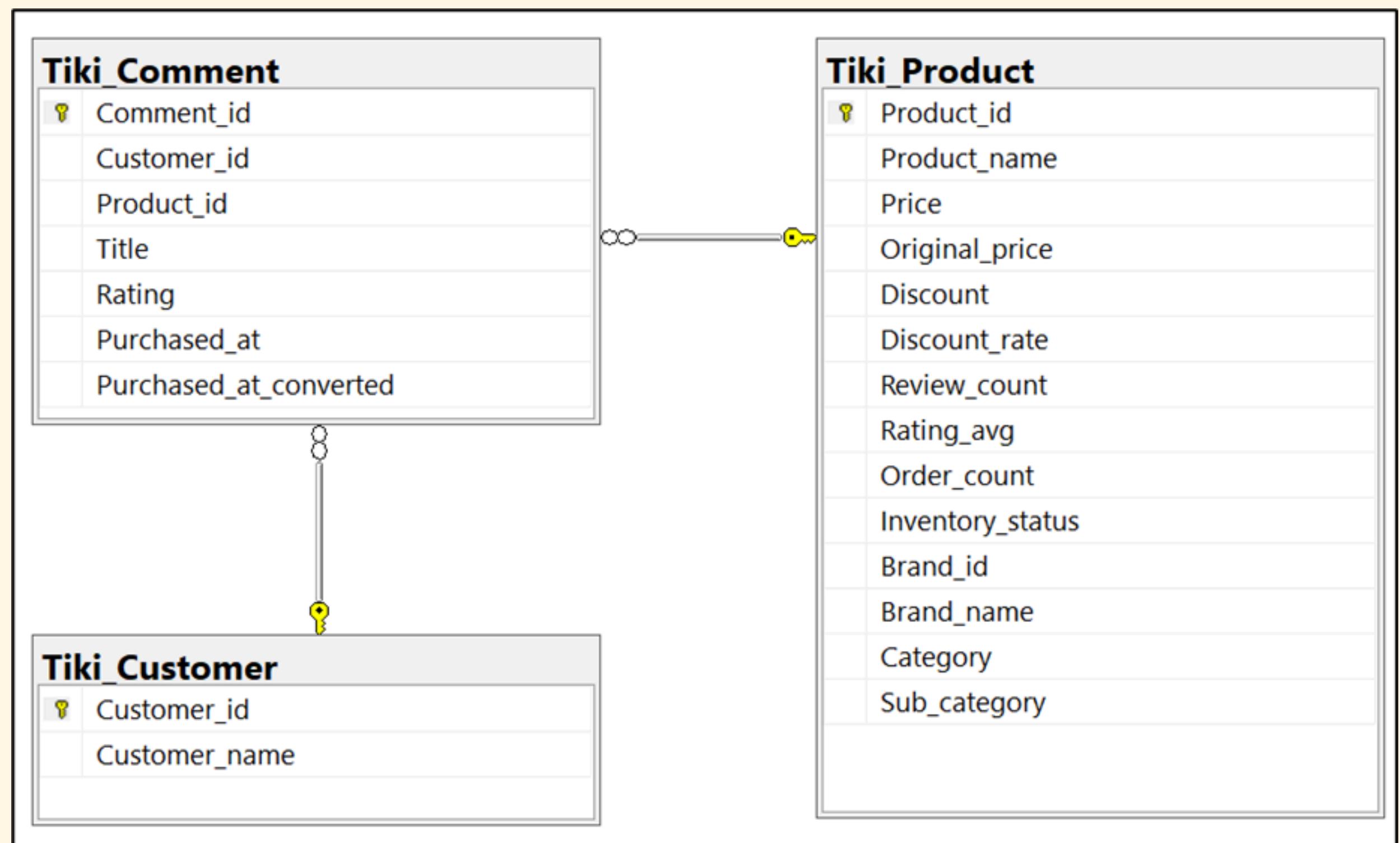


R1+R2

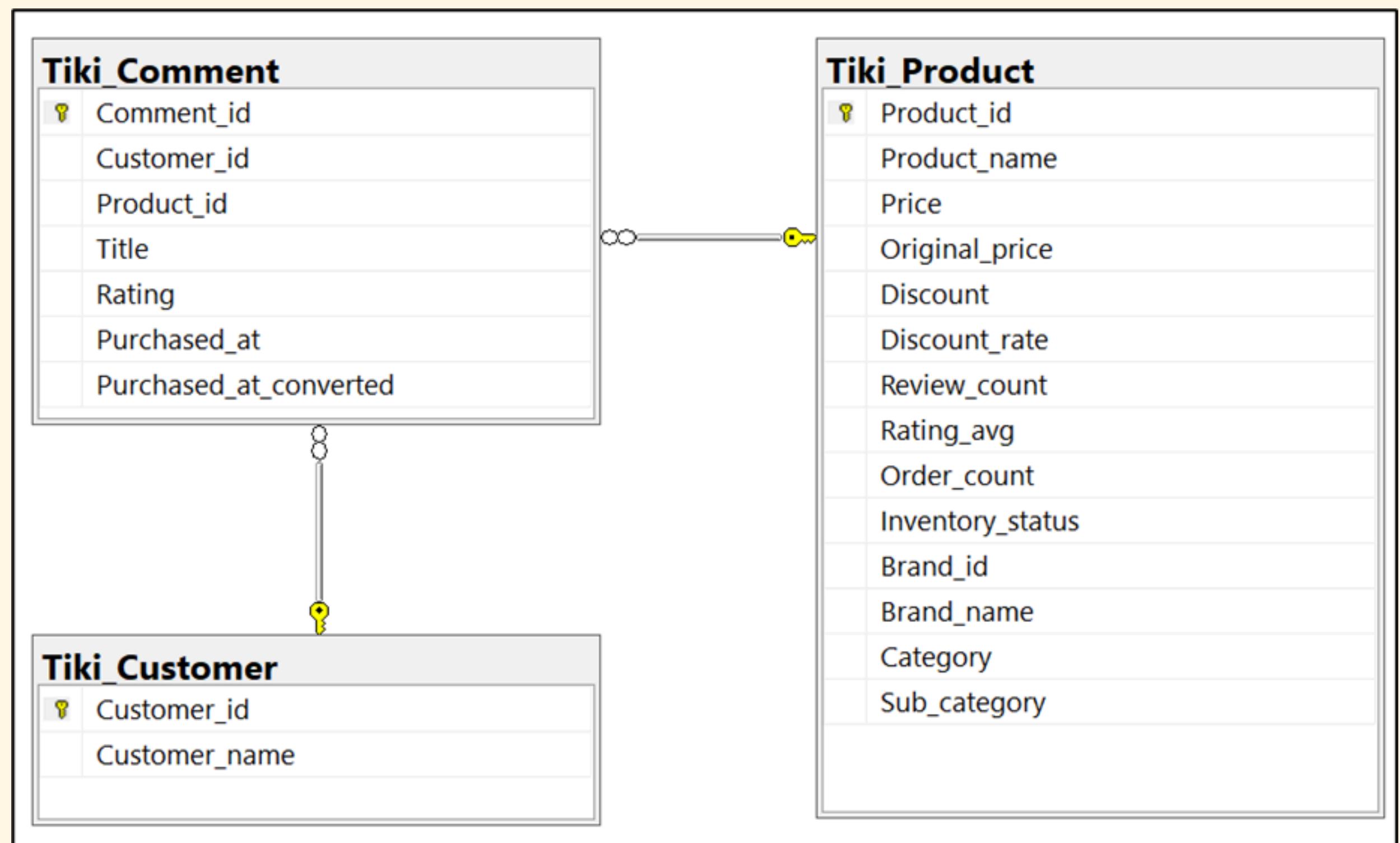
Quy trình thu thập dữ liệu:

- Xác định các thông tin cần thiết có thể thu thập và tạo cơ sở dữ liệu ban đầu.
- Thu thập dữ liệu
- Kết nối và đưa dữ liệu trực tiếp vào cơ sở dữ liệu.

R1+R2



R1+R2



R1+R2



https://tiki.vn/nha-cua-doi-song/c1883

Đồ thờ cúng

Dụng cụ & Thiết bị tiện ích

TIKI DUY NHẤT THÁNG 12

* ĐÓN MỪNG GIÁNG SINH RINH NGÀN DEAL ĐỈNH

COUPON THẢ GA QUÀ TẶNG NGẬP TRẦN GIAO NHANH 2H

Nestlé NESCAFE 3-in-1 MILO

MUA NGAY

DÚA SẤY DẺO Tài trợ

LOCKNLOCK Bộ Cây Lau Nhà Xoay Tay Compact Spin Mop Lock&Lock ETM466: Thùng Giặt + Cây +...

LOCKNLOCK Bình Giữ Nhiệt Lock&Lock Energetic One-Touch Tumbler LHC3249 - 550ML

SUMIKA Giường xếp gấp gọn di động SUMIKA 386, khung thép, có bánh xe

LOCKNLOCK Bình giữ nhiệt LocknLock Metro Drive LHC4277S 650ml - Kèm ống hút và cọ rửa

GREENHOME SIÊU THỊ LÂM VƯỜN

MƯỜNG ĐÔNG DINH LƯỢNG PHÂN BÓN DẶNG BỘT

Tin mới

Trợ lý

TOP DEAL FREESHIP XTRA CHÍNH HÃNG

FREESHIP XTRA CHÍNH HÃNG

FREESHIP XTRA CHÍNH HÃNG

FREESHIP XTRA CHÍNH HÃNG

8 Màu

Giao thứ 2, 23/12

10 Gram

05 Gram

02 Gram

01 Gram

100% naturel.de kraft btun Contenance:33 litres.

100%Ecographi.

Ne le jetez pas.

1 l peut servir plusieurs fois.

Giao thứ 2, 23/12

Giao thứ 3, 24/12

10 Gram

05 Gram

02 Gram

01 Gram

R1+R2



nhiều hơn cùng **FREESHIP XTRA**

Tìm kiếm

ẹp nhà cửa sách thể thao vắt nước chanh

% nếu hàng già | 30 ngày đổi trả | Giao nhanh 2h | Giá siêu rẻ

> Dụng cụ lau nhà > Chổi, cây lau nhà > Bộ lau nhà > Bộ Cây Lau

FREESHIP XTRA 30 NGÀY ĐỔI TRẢ CHÍNH HÃNG

Thương hiệu: LocknLock

Bộ Cây Lau Nhà Xoay Tay Compact Spin Mop Lock&Lock ETM466: Thùng Giặt + Cây + Bông Lau

4.6 ★★★★★ (1339) | Đã bán 8k

539.000đ -48% 1.034.000đ ⓘ

Giá sau áp dụng mã khuyến mãi

Giảm 30.000đ từ mã khuyến mãi của nhà bán

Thông tin vận chuyển

Giao đến Q. Hàm Cá, P. Hòn Rơm, TP. Bà Rịa

➡ Giao Thứ Hai

Trước 19h, 23/12: 10.900đ 110.900đ

Trợ lý

Đổi

Tin mới

Network

Filter

Invert

More filters

All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other

Name

- enrolls
- suggestion?trackity_id=...
- me
- token
- info
- combo?seller_id=1&pla...
- 3480627?platform=web...
- reviews?limit=5&includ...
- review-analysis?product...
- reviews?product_id=34...
- coupon?platform=desk...
- 3480627?seller_id=1
- seller?seller_id=1&mi...

41 / 258 requests | 79.4 kB / 2

Console Issues +

Default levels 99+ 4 hidden

user path: /bo-cay-lau-nha-xoay-tay-compact-spin-mop-lock-lock-etm466-thung-giat-cay-bong-lau-p3480627.html amc-core.min.js:1

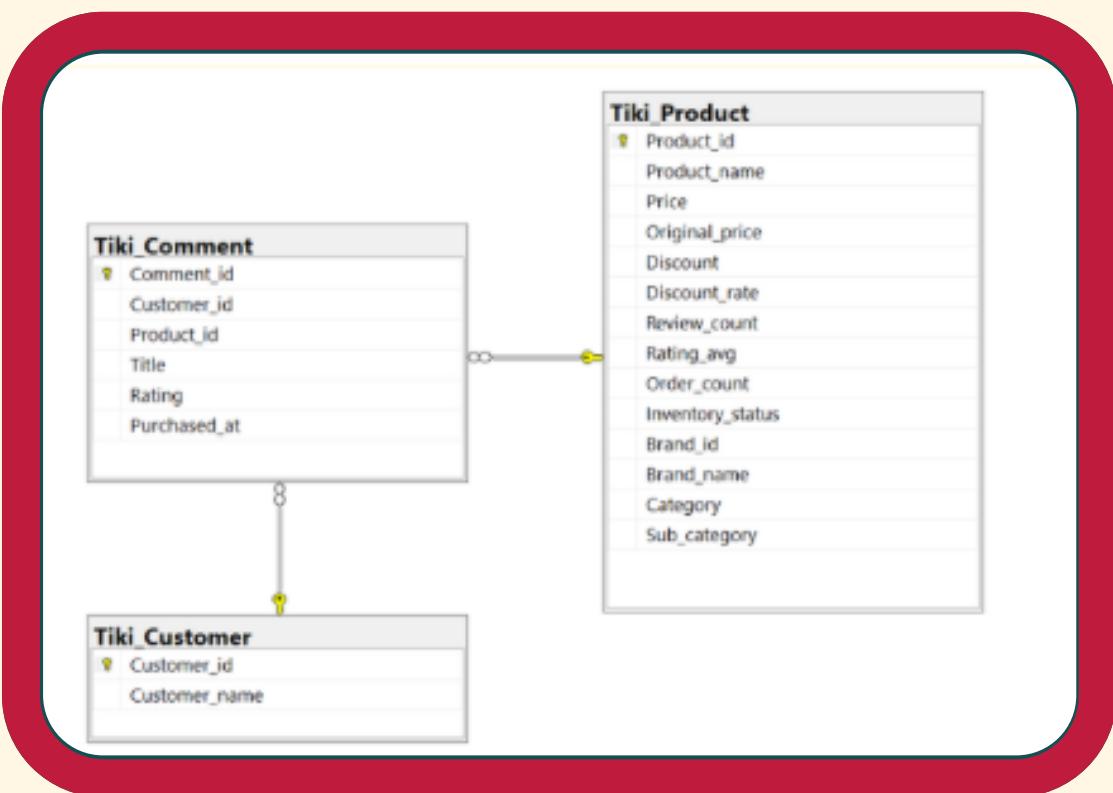
check path: false amc-core.min.js:1

fbq pageview VM875:1

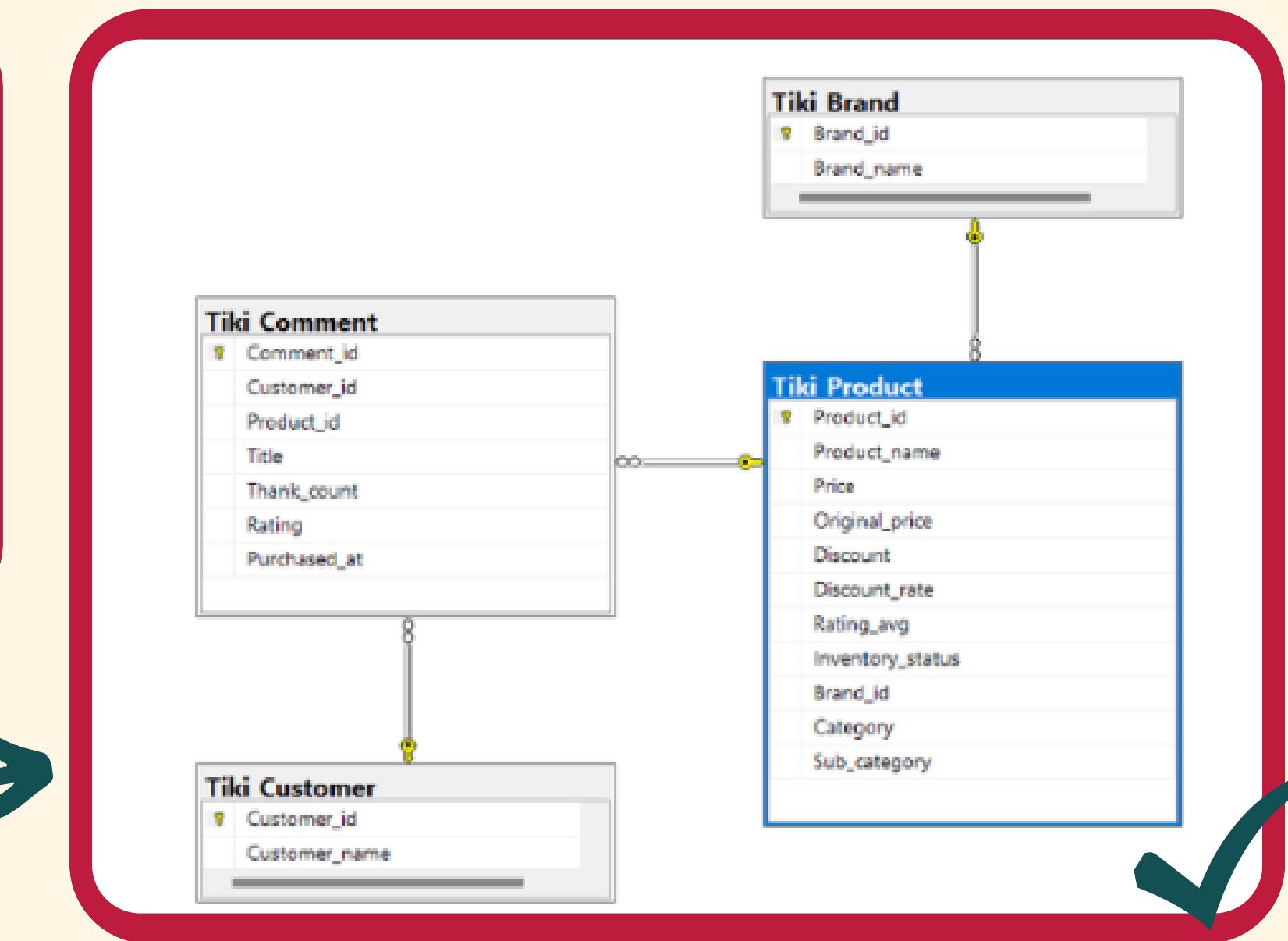
669864700220301428 get ckid persist.js:176

R3

Phân chia cấu trúc cho cơ sở dữ liệu và tiền xử lý



Phân chia cấu trúc cho
cơ sở dữ liệu và tiền
xử lý



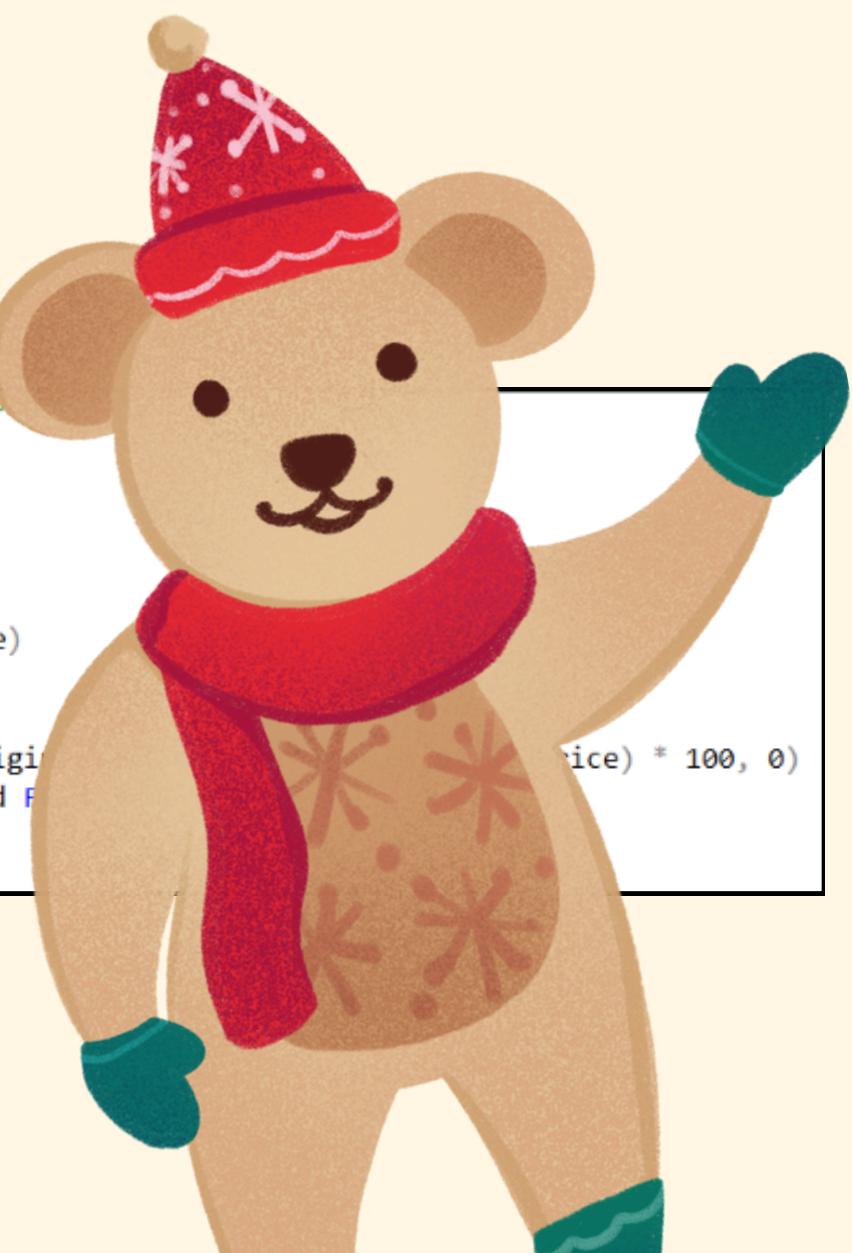
CSDL hoàn chỉnh

Phương Anh

R3

Tiền xử lý

- Xóa các cột không cần thiết, xử lý NULL
- Tạo 1 số Procedure
- Tạo 1 số Trigger



```
--Xóa các giá trị null
CREATE PROCEDURE DeleteRowsWithNull
AS
BEGIN
    DELETE FROM Tiki_Comment
    WHERE Title IS NULL
        OR Rating IS NULL
        OR Purchased_at IS NULL;
END;
```

```
--Xóa bình luận sau khi xóa sản phẩm
CREATE TRIGGER DeleteCommentsAfterProductDelete
ON Tiki_Product
AFTER DELETE
AS
BEGIN
    DELETE FROM Tiki_Comment
    WHERE Product_id IN (SELECT Product_id FROM deleted)
END
```

```
--Cập nhật lại % giảm giá sau khi
--CREATE TRIGGER UpdateDiscountRate
ON Tiki_Product
AFTER UPDATE
AS
BEGIN
    IF UPDATE (Price) OR UPDATE (Discount_rate)
    BEGIN
        UPDATE Tiki_Product
        SET Discount_rate = (SELECT (1 - (Price / Old_Price)) * 100, 0)
        WHERE Product_id = (SELECT Product_id FROM deleted)
    END;
END;
```

R4

*Sao lưu và khôi
phục dữ liệu*



Loại và thời gian backup

Nhóm sử dụng 3 loại:

- Full Backup: chủ nhật hàng tuần lúc 0 giờ.
- Differential Backup: mỗi tối lúc 0 giờ
- Transaction Log Backup: mỗi 15 phút trong ngày

Tên	Ngày sửa đổi	Loại
Differential_Backup.bak	07/11/2024 11:10 CH	Tệp BAK
DifferentialBackup.bak	08/11/2024 12:00 SA	Tệp BAK
Full_Backup.bak	07/11/2024 11:03 CH	Tệp BAK
Log_Backup.bak	07/11/2024 11:14 CH	Tệp BAK
TransactionLogBackup.trn	08/11/2024 12:30 SA	Tệp TRN

```
BACKUP DATABASE DATA_For_ML  
TO DISK = 'C:\DATA_Backup\Full_Backup.bak';  
  
BACKUP DATABASE DATA_For_ML TO DISK = 'C:\DATA_Bac  
WITH DIFFERENTIAL;  
  
BACKUP LOG DATA_For_ML TO DISK = 'C:\DATA_Backup\L
```

```
-- Lịch chạy Full Backup: Mỗi tuần một lần (chủ nhật 12 giờ sáng)  
EXEC sp_add_jobschedule  
    @job_name = N'Auto_Backup_DATA_For_ML',  
    @name = N'Weekly Full Backup',  
    @freq_type = 8, -- Chạy hàng tuần  
    @freq_interval = 1, -- Chủ nhật  
    @active_start_time = 000000; -- 12:00 tối  
GO  
  
-- Lịch chạy Differential Backup: Mỗi ngày vào 12 giờ tối  
EXEC sp_add_jobschedule  
    @job_name = N'Auto_Backup_DATA_For_ML',  
    @name = N'Daily Differential Backup',  
    @freq_type = 4, -- Chạy hàng ngày  
    @freq_interval = 1,  
    @active_start_time = 000000; -- 12:00 tối  
GO  
  
-- Lịch chạy Log Backup: Mỗi 15 phút  
EXEC sp_add_jobschedule  
    @job_name = N'Auto_Backup_DATA_For_ML',  
    @name = N'15-Minute Log Backup',  
    @freq_type = 4, -- Lặp lại hàng ngày  
    @freq_interval = 1, -- Chạy mỗi ngày  
    @freq_subday_type = 4, -- Lặp lại theo phút  
    @freq_subday_interval = 15, -- Mỗi 15 phút  
    @active_start_time = 000000, -- Bắt đầu từ 12:00 sáng  
    @active_end_time = 235959; -- Kết thúc lúc 11:59 tối  
GO
```

Lưu trữ và quản lý backup

Lưu trữ:

- Nội bộ
- Google Drive

Quản lý:

- Xác định chiến lược backup
- Kiểm tra toàn vẹn của backup
- Giám sát backup qua Mail
- Sao lưu tới thiết bị ngoài
- Kiểm tra lịch sử backup

Command:

```
DECLARE @message NVARCHAR(MAX);
SET @message = "";

SELECT @message += 'Database: ' + name + CHAR(10) +
    'Last Full Backup: ' + ISNULL(CONVERT(VARCHAR, MAX(backup_fini
FROM msdb.dbo.backupset
WHERE type = 'D'
GROUP BY name;

EXEC msdb.dbo.sp_send_dbmail
    @profile_name = 'BackupAlerts',
    @recipients = 'phanxuanhaianh@gmail.com',
    @subject = 'Weekly Backup Report',
    @body = @message;
```

```
SELECT
    database_name,
    backup_start_date,
    backup_finish_date,
    CASE
        WHEN type = 'D' THEN 'Full Backup'
        WHEN type = 'I' THEN 'Differential Backup'
        WHEN type = 'L' THEN 'Transaction Log Backup'
    END AS BackupType,
    physical_device_name
FROM msdb.dbo.backupset
INNER JOIN msdb.dbo.backupmediafamily
ON backupset.media_set_id = backupmediafamily.media_set_id
ORDER BY backup_finish_date DESC;
```

2024-11-10 09:15:00.000	2024-11-10 09:15:00.000	Transaction Log Backup	C:\DATA_Backup\Tr
2024-11-10 09:12:09.000	2024-11-10 09:12:09.000	Transaction Log Backup	C:\DATA_Backup\Tr
2024-11-10 00:15:00.000	2024-11-10 00:15:00.000	Transaction Log Backup	C:\DATA_Backup\Tr
2024-11-10 00:00:01.000	2024-11-10 00:00:01.000	Transaction Log Backup	C:\DATA_Backup\Tr
2024-11-10 00:00:01.000	2024-11-10 00:00:01.000	Full Backup	C:\DATA_Backup\Fu
2024-11-10 00:00:01.000	2024-11-10 00:00:01.000	Differential Backup	C:\DATA_Backup\Di
2024-11-09 23:45:01.000	2024-11-09 23:45:01.000	Transaction Log Backup	C:\DATA_Backup\Tr

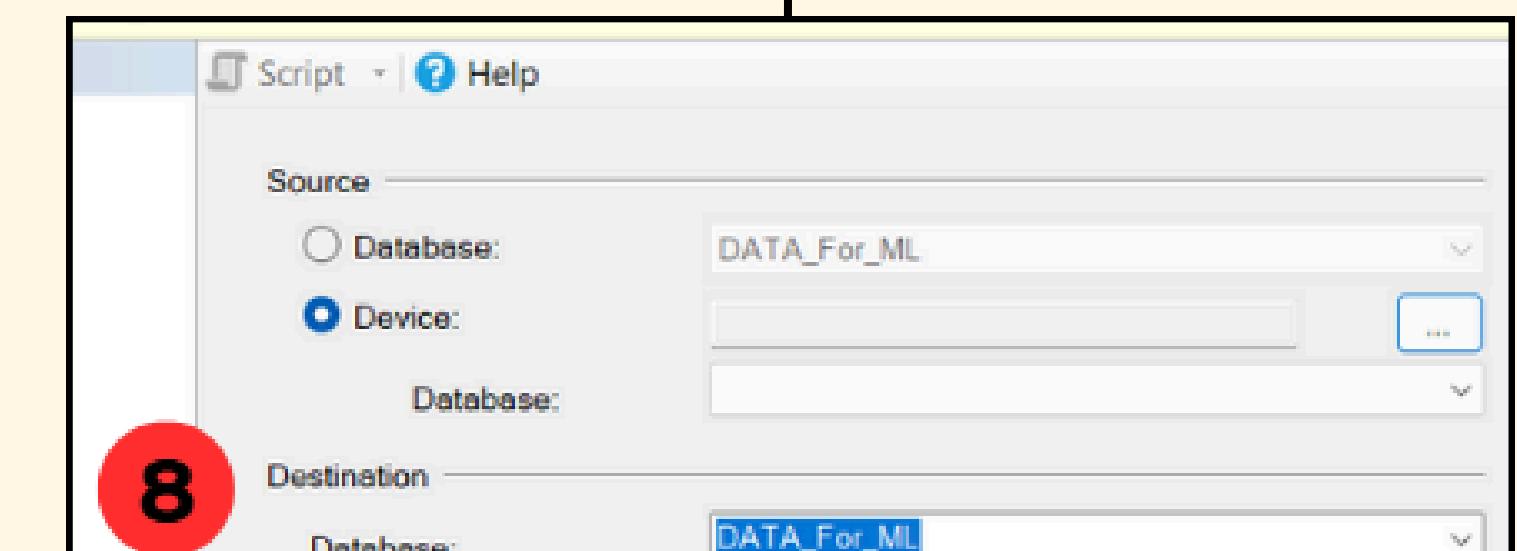
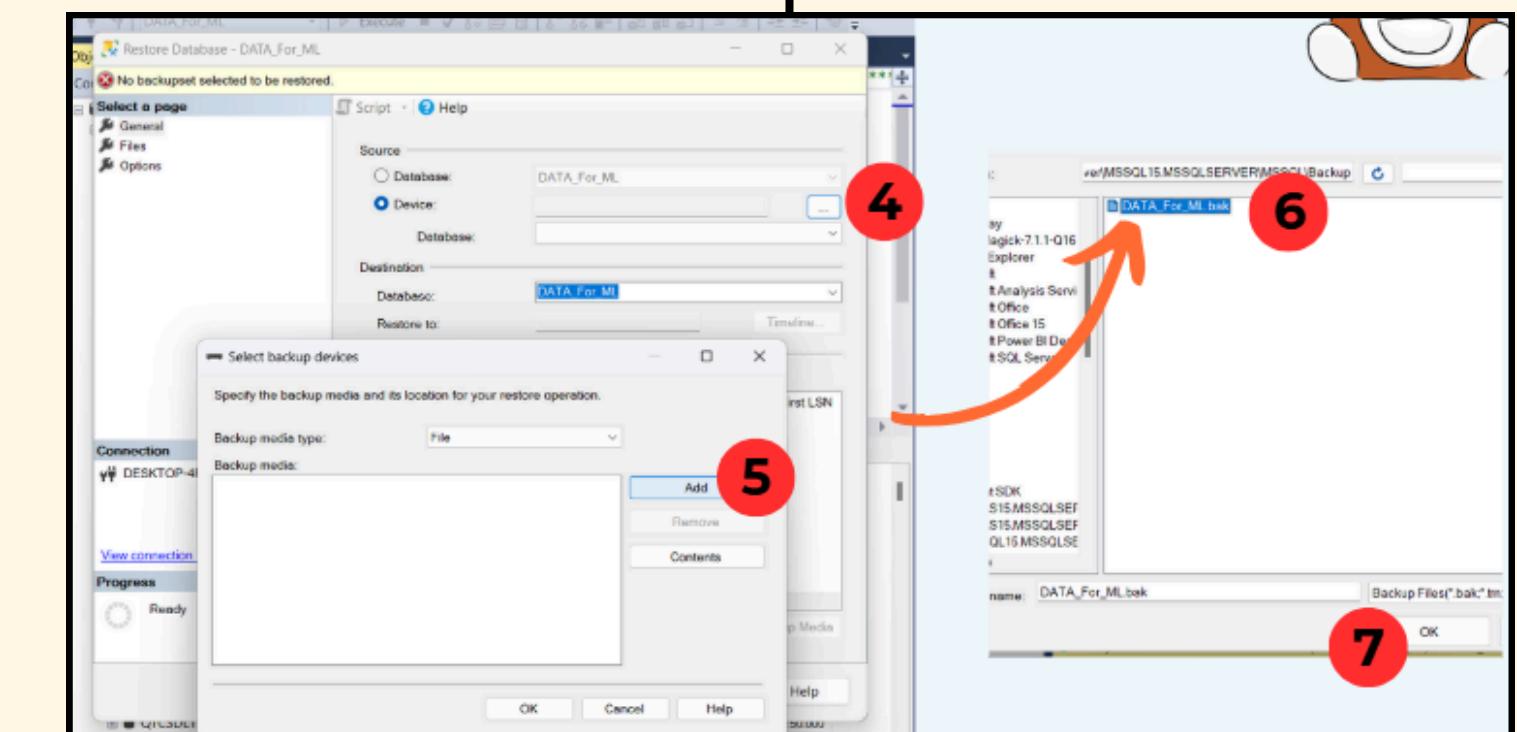
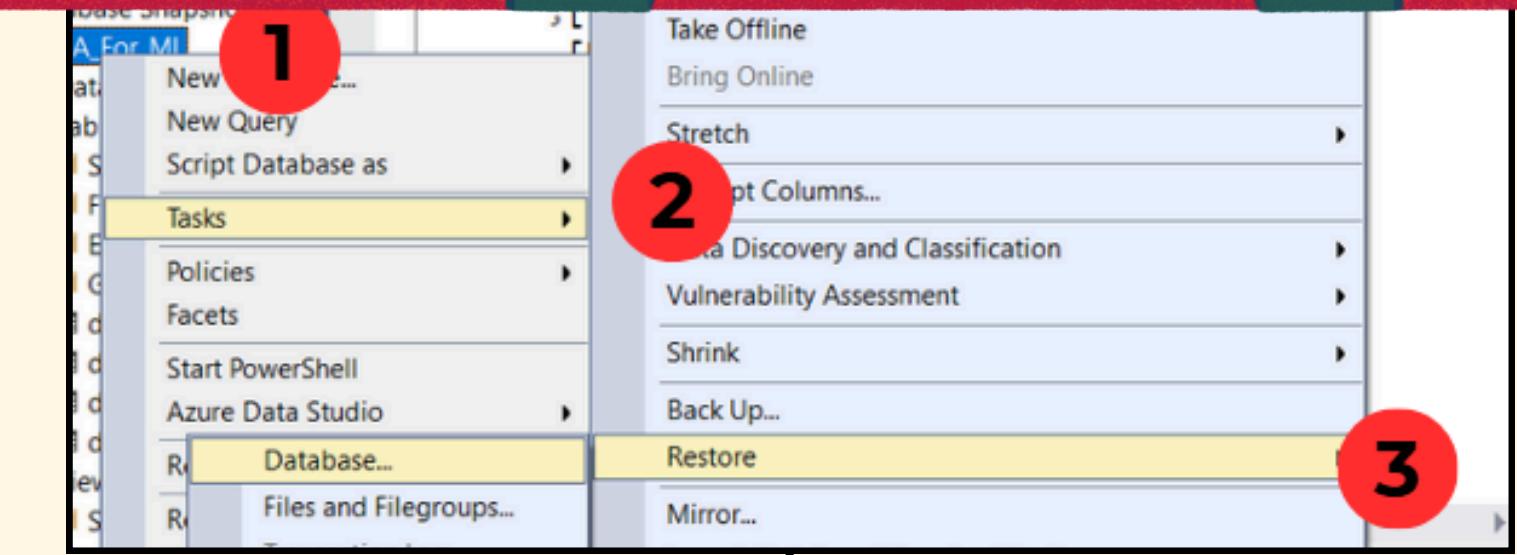
Phục hồi dữ liệu backup

B1: Đảm bảo không có ai đang truy cập vào cơ sở dữ liệu

B2: Tiến hành phục hồi từ bản sao lưu

B3: Đặt cơ sở dữ liệu về chế độ nhiều người dùng

B4: Thực hiện các truy vấn kiểm tra



R5

*Phân quyền trên
cơ sở dữ liệu*





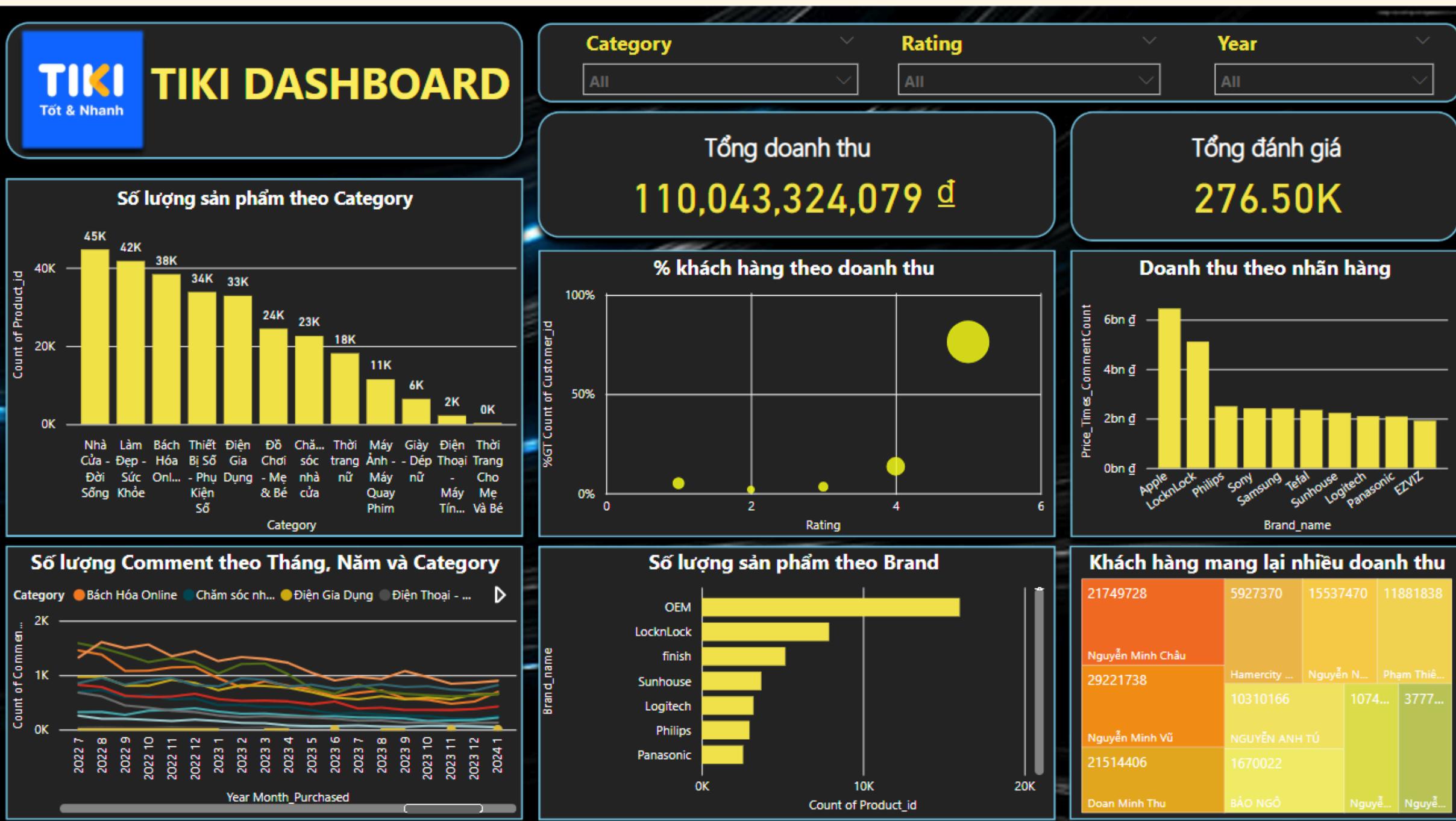
R6

Trực quan hóa dữ liệu

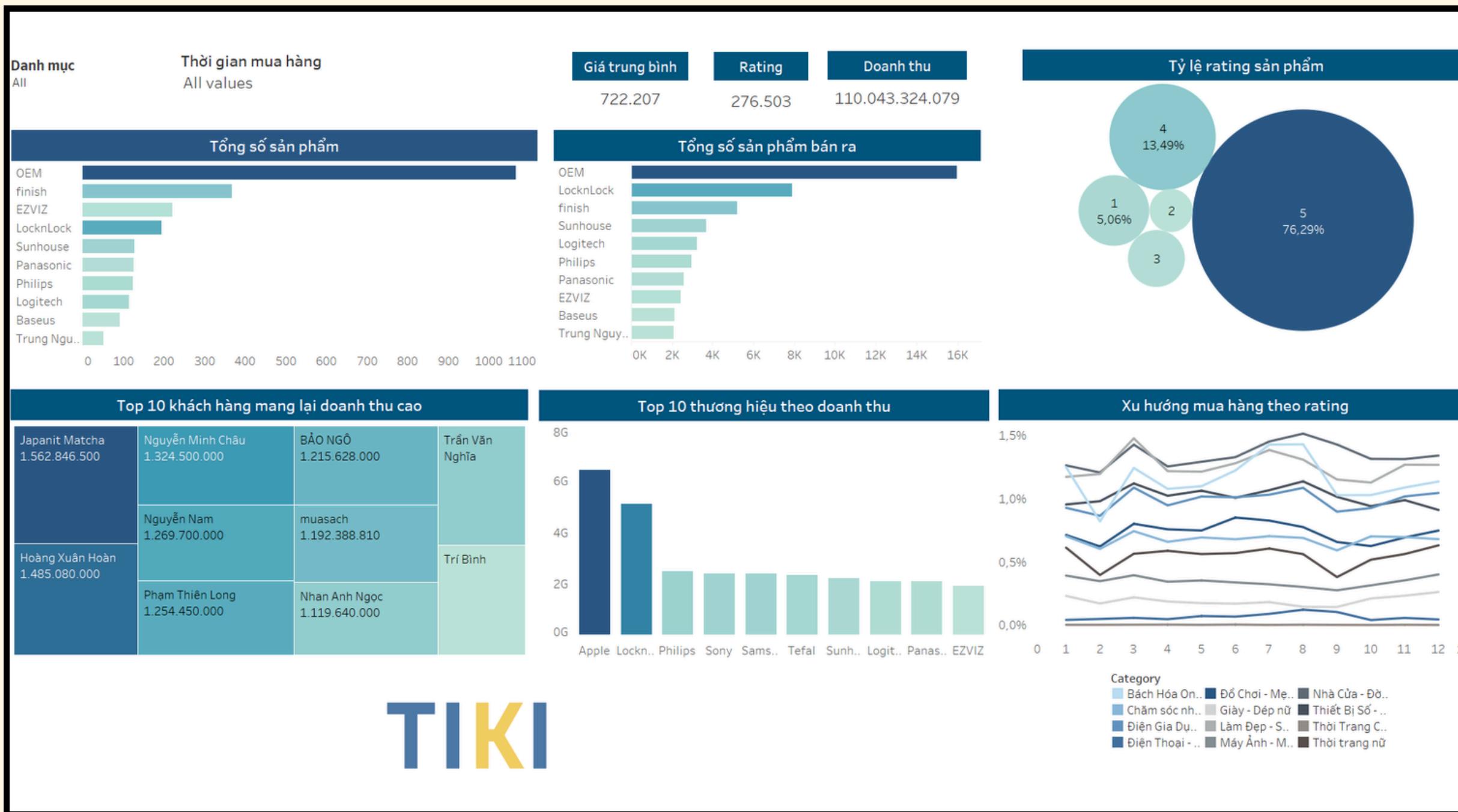


Phương Anh

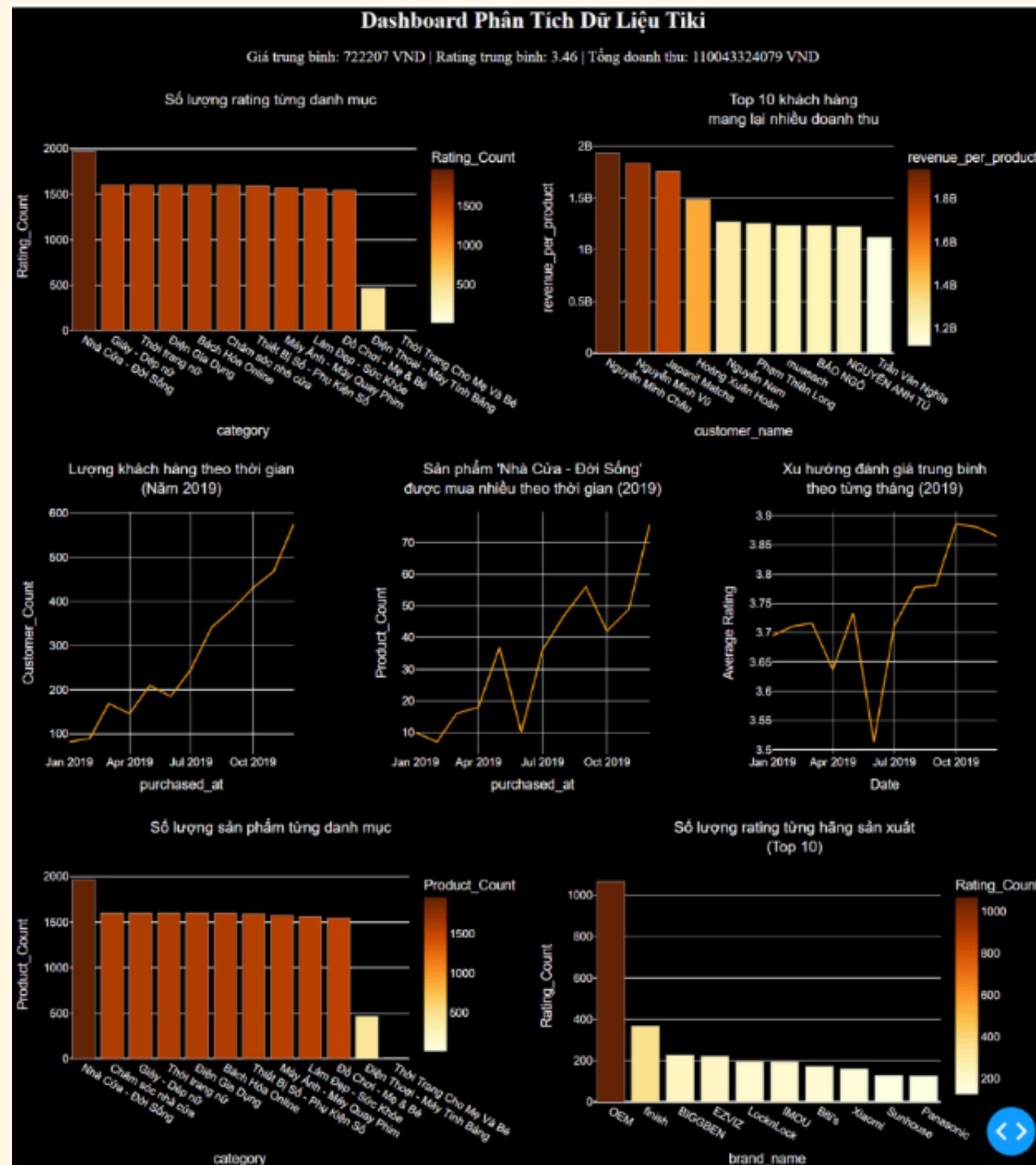
Bằng PowerBI



Bằng PowerBI



Bằng Python





R7

Mô hình

Machine Learning

Kết Nối Cơ Sở Dữ Liệu:

- Kết nối đến cơ sở dữ liệu SQL Server để lấy thông tin sản phẩm.
- Sử dụng thư viện pyodbc để thực hiện truy vấn dữ liệu.

```
server = 'DESKTOP-77HM0LQ'
database = 'DATA_For_ML'
username = 'sa'
password = 'taolakhaiden0285'

connection_String = f'DRIVER={{SQL Server}}.
```

```
try:
    # Kết nối với database và in kết quả 5 sản phẩm đầu tiên của bảng Product
    conn = pyodbc.connect(connection_String)
    query = 'SELECT * FROM Tiki_Product'

    df_sanpham = pd.read_sql(query, conn)
    print(df_sanpham.head(5))

except pyodbc.Error as e:
    print(f'Error: {e}')
finally:
    if conn:
        conn.close()
```



Tiền xử lý Dữ Liệu :

- Tạo cột combinedFeatures bằng cách ghép tên sản phẩm và giá.
- Sử dụng hàm apply() để xử lý từng dòng

```
# Tạo đặc trưng cho dữ liệu :  
features = ['Product_name', 'Price']  
def combineFeatures(row):  
    return str(row['Price']) + " " + str(row['Product_name'])  
  
df_sanpham['combinedFeatures'] = df_sanpham.apply(combineFeatures, axis=1)  
print(df_sanpham['combinedFeatures'].head())
```

0	289000.0	Tai nghe chụp tai Logitech H150 - 2 j...
1	219000.0	Chuột không dây Logitech M185 - Hàng ...
2	265000.0	Chuột Không Dây Logitech M187 - Hàng ...
3	241000.0	Loa Vi Tính SoundMax A-130/2.0 6W - ...
4	480000.0	Combo chuột phím không dây Logitech M...

Hàm Gợi Ý Sản Phẩm

1. Sử dụng thư viện fuzzywuzzy để xác định sản phẩm trong dữ liệu có tên gần giống với đầu vào nhất.
2. Dùng cosine_similarity để tính toán mức độ tương đồng giữa sản phẩm được tìm thấy và các sản phẩm khác.

```
def get_similar_products(Product_name):  
    ketqua = []  
    # Tìm kiếm sản phẩm gần giống  
    Product_name_list = df_sanpham['Product_name'].tolist()  
    best_match = process.extractOne(Product_name, Product_name_list)  
    if best_match:  
        Product_name = best_match[0]  
        product_id = df_sanpham[df_sanpham['Product_name'] == Product_name]['Product_id'].values[0]  
        indexproductid = df_sanpham[df_sanpham['Product_id'] == product_id].index[0]  
  
        similarProduct = list(enumerate(similar[indexproductid]))  
  
        sortedSimilarProduct = sorted(similarProduct, key=lambda x: x[1], reverse=True)
```

```
def lay_ten(index):  
    return (df_sanpham[df_sanpham.index == index]['Product_name'].values[0])  
  
for i in range(1, number + 1):  
    ketqua.append(lay_ten(sortedSimilarProduct[i][0]))  
  
result = "Sản phẩm gợi ý:\n"  
for product in ketqua:  
    result += product + "\n"  
return result  
else:  
    return "Không tìm thấy sản phẩm"
```

Gợi Ý Sản Phẩm Dựa Trên Đánh Giá Người Dùng

1. Truy xuất danh mục (Category) của sản phẩm được chọn.
2. Lấy danh sách các sản phẩm trong cùng danh mục, ngoại trừ sản phẩm đầu vào.
3. Chọn ra n sản phẩm đầu tiên từ danh mục này làm gợi ý.

```
def collaborative_filtering(customer_id):
    # Đọc dữ liệu từ cơ sở dữ liệu
    query = 'SELECT * FROM Ratings'
    ratings = pd.read_sql(query, conn)

    # Tạo ma trận đánh giá trung bình
    user_item_matrix = ratings.pivot(index='Customer_id', columns='Product_id', values='Rating_avg')

    # Tạo mô hình nearest neighbors
    model = NearestNeighbors(n_neighbors=5, algorithm='brute', metric='cosine')
    model.fit(user_item_matrix.fillna(0))

    # Tìm kiếm sản phẩm gợi ý cho khách hàng
    distances, indices = model.kneighbors(user_item_matrix.iloc[customer_id-1].values.reshape(1, -1), n_neighbors=5)
```

```
# Lấy danh sách sản phẩm gợi ý
# Lấy danh sách sản phẩm gợi ý
recommended_products = []
for index in indices[0]:
    product_id = user_item_matrix.columns[index]
    product_name = df_sanpham[df_sanpham['Product_id'] == product_id]['Product_name'].values[0]
    recommended_products.append(product_name)

# Trả về danh sách sản phẩm gợi ý
result = "Sản phẩm gợi ý:\n"
for product in recommended_products:
    result += product + "\n"
return result
```

Gợi Ý Sản Phẩm Dựa Trên Đánh Giá Người Dùng

1. Truy xuất danh mục (Category) của sản phẩm được chọn.
2. Lấy danh sách các sản phẩm trong cùng danh mục, ngoại trừ sản phẩm đầu vào.
3. Chọn ra n sản phẩm đầu tiên từ danh mục này làm gợi ý.

```
def collaborative_filtering(customer_id):
    # Đọc dữ liệu từ cơ sở dữ liệu
    query = 'SELECT * FROM Ratings'
    ratings = pd.read_sql(query, conn)

    # Tạo ma trận đánh giá trung bình
    user_item_matrix = ratings.pivot(index='Customer_id', columns='Product_id', values='Rating_avg')

    # Tạo mô hình nearest neighbors
    model = NearestNeighbors(n_neighbors=5, algorithm='brute', metric='cosine')
    model.fit(user_item_matrix.fillna(0))

    # Tìm kiếm sản phẩm gợi ý cho khách hàng
    distances, indices = model.kneighbors(user_item_matrix.iloc[customer_id-1].values.reshape(1, -1), n_neighbors=5)
```

```
# Lấy danh sách sản phẩm gợi ý
# Lấy danh sách sản phẩm gợi ý
recommended_products = []
for index in indices[0]:
    product_id = user_item_matrix.columns[index]
    product_name = df_sanpham[df_sanpham['Product_id'] == product_id]['Product_name'].values[0]
    recommended_products.append(product_name)

# Trả về danh sách sản phẩm gợi ý
result = "Sản phẩm gợi ý:\n"
for product in recommended_products:
    result += product + "\n"
return result
```

Qiao diện tương tác

← ⌂ ⓘ 127.0.0.1:7860 ⌂ ☆

Import favorites | yuptude | Booking.com | Lenovo Support | Lenovo | McAfee

Gợi ý sản phẩm

Nhập tên sản phẩm để nhận gợi ý

Nhập tên sản phẩm

Clear Submit

Kết quả

Sản phẩm gợi ý:

- Tai nghe chụp tai Logitech H150 - 2 jack 3.5mm, Mic khử giảm tiếng ồn, âm thanh nổi - Hàng chính hãng
- Chuột Không Dây Logitech M187 - Hàng chính hãng
- Loa Vi Tính SoundMax A-130/2.0 6W - Hàng Chính Hãng
- Combo chuột phím không dây Logitech MK240 - Hàng chính hãng
- Tai Nghe Nhét Tai Sony Fontopia MDR-E9LP - Hàng Chính Hãng

Flag



Thank you

